UNIX* System V
Release 4.0
Version 3
Tips and Troubleshooting

Order Number: 465836-002

Intel Corporation
3065 Bowers Avenue
Santa Clara, California 95051

The information in this document is subject to change without notice.

Intel Corporation makes no warranty of any kind with regard to this material, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. Intel Corporation assumes no responsibility for any errors that may appear in this document. Intel Corporation makes no commitment to update or to keep current the information contained in this document.

Intel Corporation assumes no responsibility for the use of any circuitry other than the circuitry embodied in an Intel product. No other circuit patent licenses are implied.

Intel software products are copyrighted by and shall remain the property of Intel Corporation. Use, duplication or disclosure is subject to restrictions stated in Intel's Software License, or as defined in ASPR 7-104.9(a)(9).

No part of this publication may be reproduced or transmitted in any form or by any means – graphic, electronic, electrical, mechanical, or chemical, including photocopying, recording in any medium, taping, by any computer or information storage and retrieval systems, etc., without prior permission in writing from AT&T and Intel Corporation.

The following are trademarks of Intel Corporation and its affiliates and may be used only to identify Intel products:

Above
ACE51
ACE96
ACE186
ACE196
ACE960
Action Media
BITBUS
COMMputer
CREDIT
Data Pipeline
DVI
ETOX
FaxBACK
Genius
Δ
i
i®
i386
i387
i486
i750
i860

I²ICE
ICE
iCEL
ICEVIEW
iCS
iDBP
iDIS
iLBX
iMDDX
iMMX
Inboard
Insite
Intel®
int$_e$l®
Intel386
Intel387
Intel486
int$_e$lBOS
Intel Certified
Intelevision
int$_e$ligent Identifier
int$_e$ligent Programming

Intellec®
Intellink
iOSP
iPAT
iPDS
iPSC®
iRMK
iRMX®
iSBC®
iSBX
iSDM
iSXM
Library Manager
MAPNET
MCS®
Megachassis
MICROMAINFRAME
MULTI CHANNEL
MULTIMODULE
MultiSERVER
ONCE
OpenNET

OTP
PRO750
PROMPT
Promware
QUEST
QueX
Quick-Erase
Quick-Pulse Programming
Ripplemode
RMX/80
RUPI
Seamless
SLD
SugarCube
Tech Direct
ToolTALK
UPI
Visual Edge
VLSiCEL
ZapCode
4-SITE

XENIX, MS-DOS, Multiplan, and Microsoft are trademarks of Microsoft Corporation. UNIX and OPEN LOOK are registered trademarks of UNIX System Laboratories, Inc. in the U.S.A. and other countries. Teletype, WE, and Documenter's Workbench are registered trademarks of AT&T. ETHERNET is a trademark of Xerox Corporation. Centronics is a trademark of Centronics Data Computer Corporation. Chassis Trak is a trademark of General Devices Company, Inc. VAX, VMS, and PDP are trademarks of Digital Equipment Corporation. Smartmodem 1200 and Hayes are trademarks of Hayes Microcomputer Products, Inc. IBM and PC AT are registered trademarks of International Business Machines. MDS is an ordering code only and is not used as a product name or trademark. MDS is a registered trademark of Mohawk Data Sciences Corporation. X Window System is a trademark of Massachusetts Institute of Technology. Crystal/Writer is a registered trademark of Syntactics Corporation. VP/ix is a registered trademark of Phoenix Technology. WEITEK is a registered trademark of MicroWay. PostScript is a registered trademark of Adobe Systems. SunOS is a trademark of Sun Microsystems; NFS and SPARC are registered trademarks of Sun Microsystems. MERGE 386 and PC-Interface are registered trademarks of LOCUS Computing. OSF, OSF/1, OSF/Motif and Motif are registered trademarks of Open Software Foundation. Motorola 68000 and Motorola 88000 are trademarks of Motorola Corporation.

© Copy ight 1991, 1990, Intel Corporation. All Rights Reserved.

REV.	REVISION	DATE
-001	Original Issue.	07/90
-002	Updated to support System V Release 4 Version 3 UNIX.	06/91

Contents

1 INTRODUCTION

2 KERNELS

3 PORT AND DEVICE CONFIGURATION

Figures and Tables

1 INTRODUCTION

Introduction

This manual contains procedures for users and administrators of the SVR4 operating system. Some of these procedures (such as rebuilding the kernel) are useful in many situations, while others (such as configuring the system for two WD7000 SCSI Host Adapter cards) are more specialized.

If what you are looking for isn't in this book, look through the SVR4 manuals for the answer, beginning with the Product Overview and Master Index, or call your service representative for assistance. As of June 1991, technical support for SVR4 UNIX (for the AT bus) will be provided by Interactive Systems Corporation (ISC). If you have any questions, call ISC at (213) 453-8649.

About This Manual

Organization

This book contains mostly short step-by-step procedures, with a few theoretical explanations when warranted.

This manual has five chapters, as follows.

1. Introduction. Contains an explanation of the manual, including notational conventions.
2. Kernels. Describes how to boot from an alternate kernel, and how to rebuild the kernel.
3. Port and Device Configuration. Explains how to configure:
   - the system for two Western Digital SCSI Host Adapter cards and two or three DPT SCSI Host Adapter cards
   - the system for more than two SCSI hard disk or tape drives
   - the serial port into the kernel
   - the system for a terminal (using Quick Terminal Setup)
   - a port monitor and port service so a terminal or dial-in modem can be connected to the serial port
   - postscript printers.
4. X Window System. Explains how to increase the number of X windows allowed, and contains information on configuring X Windows to work over a network.
5. Networking. Explains how to establish and control who has access to a system or systems via the network, and how to set up anonymous `ftp`. Troubleshooting tips are included.

Audience

Most of this manual expects you to have some technical experience as a computer user. Chapters 3 and parts of Chapter 5 are written for more experienced UNIX users and programmers. You will probably use this manual as a reference and tutorial for occasionally performed tasks.

Conventions

There are two types of conventions in this manual: notational conventions define how words should be read, and use conventions define how actions should be taken.

Notational Conventions

The notational conventions used in this manual are:

- Commands and command options, literal command examples, screen prompts, and screen displays are in `constant width`
- Commands that you type in literally, file names, directory names, and command names are in `constant width`
- Keyboard references are shown in the key graphic: ↵
- Keys together, such as Ctrl-Alt-Del means that you simultaneously press the specified keys.
- *Italic* font is used to reference parameters and titles of other manuals.

Use Conventions

The use conventions in this manual are:

- End all command lines by pressing ↵, unless otherwise noted.
- Instructions to "type" means to type the command or commands as directed and end each command line by pressing ↵.
- Instructions to "press" means to press the specified key or keys only. Do not press ↵ at the end of the action.

Long Command Lines

If an example of a command is too long to fit on the line, a backslash (\) is used to indicate that the command continues on the next line. You do not have to type the backslash. However, if you type \ and then press ⏎, the system prompts you to continue the command by displaying a right angle (>) bracket. The command line terminates when ⏎ is pressed.

2 KERNELS

Introduction

This chapter tells you how to rebuild the kernel after making changes to it, and how to boot from an alternate kernel (in case the kernel becomes unbootable).

About the System V Release 4 Kernel

The System V Release 4 (SVR4) kernel provides system services, including:

- CPU allocation time
- Memory allocation
- Process creation, communication, and termination
- File allocation
- File system maintenance operations
- Device control

The file associated with the kernel is /stand/unix. Never edit this file directly. If you change the kernel, rebuild it, as described in the next section. For example, to add a device to the system, use the sysadm interface, then rebuild the kernel.

To tune kernel parameters for performance, use the idtune command to edit related configuration files, then rebuild the kernel. Information on the idtune(1M) command is in the *System Administrator's Reference Manual.*

You can boot the system from a file in the /stand directory, as described in a later section. Also in the /stand directory is a file called unix.old, which is a copy of the previous kernel. If you change and rebuild the kernel and the system does not boot, you can boot the system from the unix.old file and fix the problem, as described in a following section.

Rebuilding the Kernel

If you make a change to the kernel (for example, you tune kernel parameters or add a device driver) you must rebuild the kernel and reboot the system before the change is reflected in system operation.

CAUTION The file associated with the kernel is `/stand/unix`. Never edit the `/stand/unix` file directly. If you are going to change the kernel, do so only by rebuilding it.

Before you modify the kernel in any way, read the discussion on kernels in the *System Administrator's Guide*.

To rebuild the kernel, do the following:

1. Make sure all user sessions are terminated.
2. Log in as root.
3. Rebuild the kernel by typing:

   ```
   /etc/conf/bin/idbuild
   ```

   The kernel rebuild can take up to 10 minutes. When the rebuild is complete, the system prompt appears.
4. When the prompt appears, shut down and reboot the system (to activate the new kernel) by typing:

   ```
   shutdown -y -g0
   ```
   Ctrl-Alt-Del

   When the system prompt returns, you can log in. The changes to the kernel will have taken effect.

Booting from an Alternate Kernel

The system may not be able to boot from the file in /stand/unix. For example, if you changed the kernel and rebuilt it, the changes may have caused the kernel to become unbootable. When you build a kernel, the previous kernel is saved in a file called /stand/unix.old. The /stand/unix.old kernel should boot unless your system has hardware problems. /stand/unix.old does not contain the changes you made before you rebuilt the kernel.

CAUTION Never change the contents of /stand/unix.old.

If you suspect your changes to the kernel made the system unbootable, you can boot the system from the old kernel. Then you can log onto the system, rectify the error that caused the kernel to fail, then rebuild the kernel to implement the desired changes. The procedure that follows tells you how to do this.

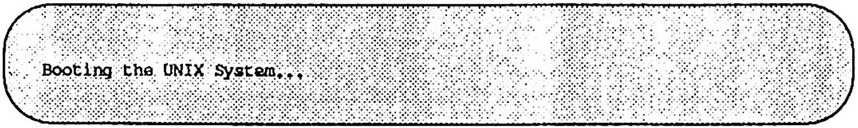
NOTE The system can boot only from a kernel that is in /stand. Regardless of the pathname typed, the kernel is assumed to be in /stand. So, even if you specify a pathname that doesn't include /stand, the directory path part of the filename is removed, and /stand is searched for the named kernel. For example, if you type /etc/conf/cf.d/unix.mine, the system tries to boot from the kernel /stand/unix.mine.

To boot from an alternate kernel, do the following:

1. Reset the system, or if it is up, shut it down and reboot it by typing:

   ```
   shutdown -y -i6 -g0
   ```

2. When the message:

   ```
   Booting the UNIX System...
   ```

   appears on the console, press the space bar.

   The system responds:

```
Enter the name of a kernel to boot:
```

3. Type the name of the file in `/stand` you wish to boot. For example, type:

   `unix.old`

   The system responds:

```
Booting the UNIX System...
```

   The specified kernel boots.

   Then the system displays:

```
The system is ready
login:
```

4. Log in as root.
5. Get in the `stand` directory.
6. Make a backup copy of the `/stand/unix.old` file. For example, type:

   `cp unix.old unix.save`

NOTE

This step is recommended because if the kernel fix you try in the next step doesn't work, you can still boot the system from the `/stand/unix.save` file. If you don't save a copy of the file and the fix doesn't work, you may be stuck with an unbootable system.

7. Do what it takes to fix the kernel; for example, run `idtune`(1M) or back out the changes you made.

8. Rebuild the kernel by typing `/etc/conf/bin/idbuild`.
9. Reboot the system.
10. If the system still does not boot, make sure there are no hardware-related problems. Try to fix the kernel again. Call your service representative if you need help.

3 PORT AND DEVICE CONFIGURATION

Introduction

This chapter describes:

- Configuring the system for two or more WD-7000 and DPT SCSI Host Adapter (controller) cards
- Configuring the system for more than two SCSI hard disk and tape drives
- Configuring the Serial Port
- Configuring a Modem (creating a port monitor and port service)
- Configuring a Terminal (using Quick Terminal Setup)
- Configuring a Postscript printer

Configuring the System for Two WD-7000 SCSI Host Adapter Cards

The following procedure tells you how to configure the system to support a second WD-7000 SCSI Host Adapter (controller) card.

WARNING

Installing controller cards involves removing the cover on the computer. When the cover is open, you are exposed to dangerous voltages. To reduce the chance of severe shock, unplug the power cords of the computer and all peripherals attached to it before opening the cover. Only qualified service technicians or system integrators should perform this task.

To configure the system for a second WD 7000 (Western Digital) FASST SCSI Host Adapter card, do the following:

1. Verify that the first WD7000 host adapter card, is configured correctly. The default configuration (IRQ 15, DACK/DRQ 6, BIOS ROM address 0xCE000, I/O address range 350 -> 353) is acceptable.

   The jumpers on the card should be set as follows (these are the default jumper settings):

   - W2: 3 to 4, 9 to 10, 13 to 14
   - W3: 1 to 2, 5 to 6, 9 to 10
   - W4: 7 to 8
   - W5: 1 to 2

2. Configure the second WD7000 host adapter card as follows (IRQ 11, DACK/DRQ 5, BIOS ROM address 0xC8000, I/O address range 330 -> 333). The jumpers on the card should be set as follows:

   - W6 and W9: in (to disable the floppy hardware)
   - W1: no jumpers (for IRQ Channel)
   - W2: 5 to 6, 11 to 12, 19 to 20 (for IRQ 11)
   - W3: 1 to 2, 7 to 8, 9 to 10 (for I/O address range 0x330 -> 0x333)
   - W4: 1 to 2, 3 to 4, 7 to 8 (for BIOS ROM address 0xC8000, which is not used).

NOTE See the hardware reference manual that came with the hardware for details on how to set jumpers.

3. Disable the BIOS on the second WD7000 by doing one of the following:
   - If jumper W98 (near Z2, lower terminator) exists, remove it. (Newer WD7000 cards have this jumper.)
   - If there is no W98 jumper (and no CR5 LED and no F1 fuse), you have an older card; remove the BIOS ROM (U60) instead.
4. Turn off power to all peripherals attached to the computer.
5. Turn off power to the computer.
6. Remove the cover of the computer.
7. Install both WD7000 Host Adapter cards in the 16-bit expansion slots.
8. Connect a SCSI cable to each WD7000.
9. Connect the other SCSI devices to the SCSI cables.
10. Make sure that each SCSI cable is properly terminated at both ends.
11. Reinstall the computer's cover.
12. Turn on power to all the hardware attached to the system.
13. Boot the system by turning on power to the computer.
14. When the system prompt displays, log in as root and edit the /etc/conf/sdevice.d/scsi file so that it looks like the following:

```
scsi  Y  1  5  1  15  350  353  0  0
scsi  Y  1  5  1  11  330  333  0  0
```

15. Rebuild the UNIX kernel by typing:

```
/etc/conf/bin/idbuild
```

You may need to remake the device nodes, depending on the how the devices are hooked up to the two SCSI buses. For example, if each SCSI bus has a WD7000 (ID 7), a Winchester Drive (ID 0), and a tape (ID 3), then:

- the system disk on BUS0 is specified (by default) as `/dev/dsk/c0t0d0s0`
- the tape drive on BUS0 is specified (by default) as `/dev/rmt/c0s0`
- the disk on BUS1 is specified (by default) as `/dev/dsk/c0t1d0s0` or 2s0
- and the tape drive on BUS1 is specified (by default) as `/dev/rmt/c0s1`

In this case, the disk and tape drives on BUS1 are using the wrong naming convention. The controller number and target ID number are wrong. The system will find the device since the minor number of the devices is correct. However, you may have problems later if you do not fix how the devices are named. For example, if you later add another disk drive to BUS0, the minor number associated with the disk drive on the second controller will be incorrect and you will have trouble accessing the desired drive.

To remake the device nodes (fix the naming problem) for the above example, do the following:

16. Get in the `/etc/conf/node.d` directory.
17. Edit the `st01` file and change the line that reads `rmt/c0s1` to `rmt/c1s0`.
18. Edit the `sd01` file and change the line that reads `dsk/c0t1d0s0` (or `2s0`) to `dsk/c1t0d0s0` (or `2s0`).

    Now, the controller number and target ID number for the devices on the second controller are correct.
19. Reboot the system.

NOTE

If you have three disk or tape drives in a system, you need to make the nodes for the third devices yourself, as described in the sections, "Configuring the System for More Than Two SCSI Hard Disks" or "Configuring the System for More Than Two SCSI Tape Drives"

For information about device nodes, see the `hd`(7), `sd01`(7), and `st01`(7) man pages in the *System Administrator's Reference Manual*.

Configuring the System for Two or Three DPT PM2012 SCSI Host Adapter Cards

The following procedure tells you how to configure the system for two or three DPT PM2012 EISA SCSI Host Adapter cards.

NOTE Up to three cards are supported.

To configure the system for two or three DPT PM2012 EISA SCSI Host Adapter cards, do the following:

WARNING Installing controller cards involves removing the cover on the computer. When the cover is open, you are exposed to dangerous voltages. To reduce the chance of severe shock, unplug the power cords of the computer and all peripherals attached to it before opening the cover. Only qualified service technicians or system integrators should perform this task.

1. Turn off power to the computer and all the peripherals attached to it.
2. Remove the computer's cover.
3. Install the DPT EISA card(s) in EISA slots 1 and 2 for two DPT's, and slots 1, 2, and 3 for three DPT's.
4. Connect a SCSI cable to each DPT card.
5. Connect the other SCSI devices to the SCSI cables.
6. Make sure that each SCSI cable is properly terminated at both ends.
7. Turn on power to all the hardware attached to the system.
8. Boot from the ECU diskette which comes with the cards and configure your hardware. The system will prompt you for input.
9. Remove the diskette then reboot your UNIX system.
10. When the system prompt displays, log in as root.
11. Edit the `/etc/conf/sdevice.d/scsi` file. When adding a second card in slot 2, change the file so it looks like this:

```
scsi    Y    1    5    1    14    1C88    1C90    0    0
scsi    Y    1    5    1    15    2C88    2C90    0    0
```

12. If you are adding a third card in slot 3, add another line to the file that looks like this:

    ```
    scsi    Y    1    5    1    12    3C88    3C90    0    0
    ```

13. If you added a third card, also do the following steps; otherwise, skip to Step 14.

    a. Edit the /etc/conf/pack.d/scsi/space.c file and add the following lines after the #ifdef SCSI_1 section:

    ```
    #ifdef SCSI_2
                ,
                SCSI_ID,          /* HA 2 SCSI identifier  */
                SCSI_2_VECT,      /* HA 2 interrupt vector */
                SCSI_2_SIOA       /* HA 2 base I/O address */
    #endif
    ```

    b. Edit the /etc/conf/cf.d/mdevice file and look for the eighth field in the scsi entry and change it from 2 to 3, so the line looks like this:

    ```
    scsi    Iocis    iHcrf    scsi    0    0    1    3    -1
    ```

14. Rebuild the kernel by typing:

    ```
    /etc/conf/bin/idbuild
    ```

15. When the system prompt returns, shut down and reboot the system by typing:

    ```
    shutdown -y -g0
    ```
    Ctrl-Alt-Del

16. Remake the device nodes, if necessary. See the previous procedure for instructions.

Configuring the System For More Than Two SCSI Hard Disks

SCSI-based systems can support more than two disks. If your system has MFM/ESDI (non-SCSI) controllers, only two hard disks are supported.

To configure the system for more than two SCSI hard disks, do the following procedure.

NOTE You can configure a maximum of 8 devices per SCSI controller (including the controller itself) into the system. There are up to 16 nodes per disk device.

CAUTION Configuring additional hard disks is not a simple task. If you do the procedure incorrectly, you can render your system inoperable and require system re-installation. Proceed with caution.

1. Log in as root.
2. Create a shell script called mkdisknodes that contains the following lines:

```
#
# Usage: mkdisknodes <disk #> <cntndn>
#
#  cntndn for controller, target, device (logical unit) numbers
#  e.g. mkdisknodes 2 c1t0d0, for disk-2
#  (3rd disk in system) using c1t0d0s*
n=$1
dev=$2
base=`expr $n  16`
cd /dev
#
# Determine the block major number for the disk driver
#
set `ls -l dsk/0s0 | sed -e "s/,.*$//"`
sd_bmajor=$5
#
# Determine the character major number for the disk driver
#
set `ls -l rdsk/0s0 | sed -e "s/,.*$//"`
sd_cmajor=$5
```

```
#
# Make the block device nodes
#
mknod dsk/${dev}s0 b $sd_bmajor $base
mknod dsk/${dev}s1 b $sd_bmajor `expr $base + 1`
mknod dsk/${dev}s2 b $sd_bmajor `expr $base + 2`
mknod dsk/${dev}s3 b $sd_bmajor `expr $base + 3`
mknod dsk/${dev}s4 b $sd_bmajor `expr $base + 4`
mknod dsk/${dev}s5 b $sd_bmajor `expr $base + 5`
mknod dsk/${dev}s6 b $sd_bmajor `expr $base + 6`
mknod dsk/${dev}sa b $sd_bmajor `expr $base + 10`
mknod dsk/${dev}sb b $sd_bmajor `expr $base + 11`
mknod dsk/${dev}sc b $sd_bmajor `expr $base + 12`
mknod dsk/${dev}sd b $sd_bmajor `expr $base + 13`
#
# Make the character (raw) device nodes
#
mknod rdsk/${dev}s0 c $sd_cmajor $base
mknod rdsk/${dev}s1 c $sd_cmajor `expr $base + 1`
mknod rdsk/${dev}s2 c $sd_cmajor `expr $base + 2`
mknod rdsk/${dev}s3 c $sd_cmajor `expr $base + 3`
mknod rdsk/${dev}s4 c $sd_cmajor `expr $base + 4`
mknod rdsk/${dev}s5 c $sd_cmajor `expr $base + 5`
mknod rdsk/${dev}s6 c $sd_cmajor `expr $base + 6`
mknod rdsk/${dev}sa c $sd_cmajor `expr $base + 10`
mknod rdsk/${dev}sb c $sd_cmajor `expr $base + 11`
mknod rdsk/${dev}sc c $sd_cmajor `expr $base + 12`
mknod rdsk/${dev}sd c $sd_cmajor `expr $base + 13`
```

This script, when executed, will create device nodes for the specified disks.

SVR4 systems already have device nodes for two hard disks. On SCSI-based systems, the device nodes for the first hard disk (disk0) are /dev/[r]dsk/c0t0d0s0, c0t0d0s1, etc. (For compatibility with non-SCSI disks, these nodes are also linked to 0s0, 0s1, etc.) The device nodes for the second hard disk (disk1) are /dev/[r]dsk/c0t1d0s0, c0t1d0s1, etc.

3. Execute the script by typing:

   ```
   mkdisknodes <disk no.> <cntndn>
   ```

   where *<cn>* is the controller number, *tn* is the target id number, and *dn* is the device (logical unit) number. For example, if your system has two SCSI host adapter cards installed, and the 3rd hard disk is on the second controller (designated as 1) and has a target ID of 7, type:

   ```
   mkdisknodes 2 c1t7d0
   ```

4. After the nodes are created, execute the `diskadd`(1M) utility, which partitions and formats the hard disk, makes the file systems, and configures the disk into the system. The format of the `diskadd` utility is:

   ```
   diskadd cntndn
   ```

   where *cntndn* is the the controller number (0-2), SCSI ID (0-6), and logical unit number (0-3).

NOTE

See `diskadd`(1M) in the *System Administrator's Reference Manual* for more information about using the `diskadd` utility.

Configuring the System for More Than Two SCSI Tape Drives

SVR4 SCSI-based systems by default have device nodes for two tape drives. The device nodes for the first tape drive (tape0) are `/dev/rmt/c0s0`, `c0s0n`, etc. The device nodes for the second tape drive (tape1) are `/dev/rmt/c0s1`, `c0s1n`, etc. SCSI systems can support more than two tape drives, but you must make the device nodes for the extra tape drives.

NOTE You can configure a maximum of 8 devices per controller (including the controller) into the system.

CAUTION Configuring additional tape drives is not a simple task. If you do the procedure incorrectly, you can render your system inoperable. If this happens, you must reinstall the system. Proceed with caution.

To configure the system for more than two SCSI tape drives, do the following:

1. Log in as root.
2. Create a shell script called `mktapenodes` that has the following lines:

```
#
# Usage: mktapenodes <tape #> <cnsn>
#
#  cnsn for controller and target-device numbers
#  e.g. mktapenodes 2 c1s1, for tape-2
#  (3rd tape in system) using c1s1
n=$1
dev=$2
base=`expr $n  16`
cd /dev/rmt
#
# Determine the character major number for the tape driver
#
set `ls -l tape | sed -e "s/,.*$//"`
st_major=$5
#
# Make the character (raw) device nodes
#
mknod ${dev} c $st_major $base
mknod ${dev}n c $st_major `expr $base + 1`
mknod ${dev}r c $st_major `expr $base + 2`
mknod ${dev}nr c $st_major `expr $base + 3`
```

This shell script, when executed, automatically creates the specified device nodes.

3. Execute the shell script. For example, to create a device node for the third tape drive, type the following:

   `mktapenodes 2` *`cnsn`*

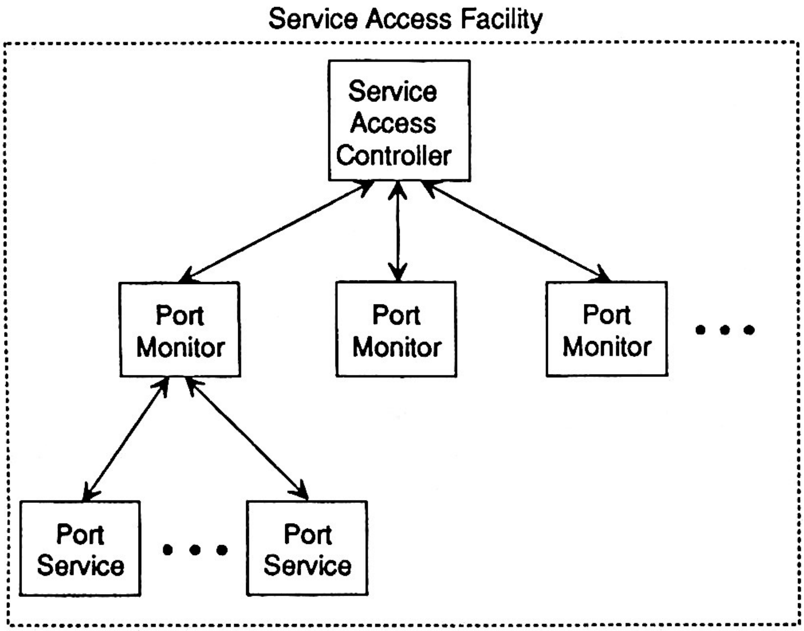
   where *cnsn* is the controller number the tape drive is connected to, and device number. The device nodes are automatically created.

Port Monitors and Port Services Overview

This section provides a brief overview of port monitors, port services, and their relationship to each other. For a detailed discussion on port monitors and port services, see the *System V Release 4 System Administrator's Guide*.

Port monitors and ports services are a part of the Service Access Facility (SAF). The SAF coordinates access to the system. The SAF consists of the Service Access Controller, port monitors and port services. Figure 3-1 shows the hierarchy of the SAF.

Figure 3-1: Service Access Facility



The Service Access Controller (SAC) is the main process and overseer of the SAF. Once the system is running, the SAC polls each of the port monitors for status. If it does not receive status from a port monitor, it assumes the port monitor has stopped running. If the port monitor should be running, the SAC tries to restart it.

A port monitor manages a port or a group of like ports. Serial ports are an example of like ports. You could have one port monitor manage all serial ports or have multiple port monitors managing smaller groups of serial ports.

A port service is the lowest level in the SAF hierarchy. It configures and monitors an individual port. For example, a port service for a serial port sets such things as baud rate and parity. There must be a port service for each port.

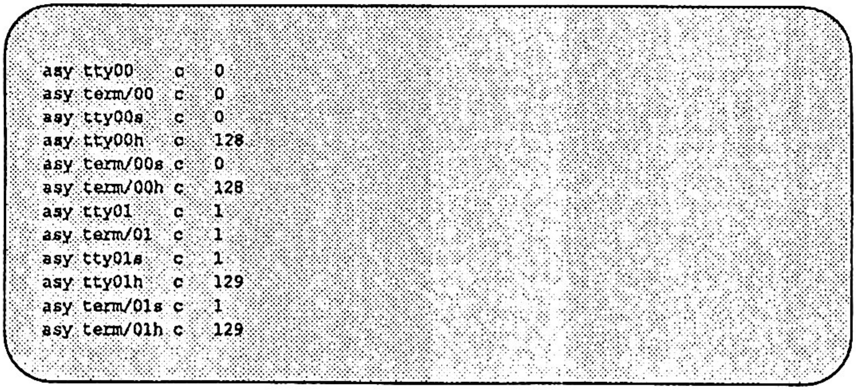
Configuring a Serial Port

Before a device can be connected to the serial port, the port must be configured into the kernel, as follows:

1. Log in as root.
2. Type `who` to make sure no one else is logged into the system. If someone is on the system, tell them to get off.
3. Edit the file called `/etc/conf/sdevice.d/asy`.
4. By default, the file contains four lines. Each line corresponds to a serial port (ports 1-4) on the mother board. Port 1 (COM1) is already configured. On the second line, change the second field from `N` to `Y`. The file should look like this:

```
asy Y  1  7  1  4  3f8  3ff  0  0
asy Y  1  7  1  3  2f8  2ff  0  0
asy N  1  7  1  2  3e8  3ef  0  0
asy N  1  7  1  5  2e8  2ef  0  0
```

5. Save the changes and end the editing session. You have just configured the channel for the serial port into the `asy` driver.
6. Now, edit the file called `/etc/conf/node.d/asy`.
7. To add device nodes for `tty01`, duplicate the first six lines of the file, then change the new lines so that the resulting file looks like this:



```
asy tty00     c   0
asy term/00   c   0
asy tty00s    c   0
asy tty00h    c   128
asy term/00s  c   0
asy term/00h  c   128
asy tty01     c   1
asy term/01   c   1
asy tty01s    c   1
asy tty01h    c   129
asy term/01s  c   1
asy term/01h  c   129
```

8. Save the changes and end the editing session.
9. Rebuild the kernel by typing:

   ```
   /etc/conf/bin/idbuild
   ```

   The kernel rebuild takes several minutes.
10. When the system prompt displays, shutdown and reboot the system by typing:

    ```
    shutdown -y -g0
    ```

    Ctrl-Alt-Del

    When the reboot is complete, the `Console login:` message appears. The com2 port is now configured into the kernel as `tty01` and is activated.

NOTE The com2 port appears in the `/dev/term` directory as the device `01`. In previous releases, the device name was expected to be in the `/dev/tty01` directory, which still exists as a file.

Now that the serial port is configured, you can configure the system so it can communicate with the device that you are going to connect to the port. To configure the system for a modem, go to the section "Configuring Modems". To configure the system for a terminal, go to the section called "Quick Terminal Setup". To configure the system for a postscript printer, go to the section called "Postscript Printer Configuration".

Configuring a Terminal (Quick Terminal Setup)

Before connecting a terminal to the serial port, you must configure the system to communicate with the terminal. The Quick Terminal Setup feature of the `sysadm` interface is easiest to use because it automatically creates a port monitor and port service.

To use Quick Terminal Setup, do the following:

1. Log in as root.
2. Invoke the `sysadm` interface.
3. Select the `Port Access Services and Monitors` menu.
4. Select the `Quick Terminal Setup` menu.
5. Select the `Add a Terminal to a Port` menu.
6. A form displays. Specify which port to use. For example, type:

   ```
   /dev/term01
   ```

   or press F2 to see the choices, then, move the cursor to the desired choice, press F2 to mark the choice, then F3 to select it.
7. The default baud rate is 9600. Select another rate if desired.
8. Press F3 to save the input.
9. Exit the `sysadm` interface.
10. Connect the terminal to the port and turn it on.
11. Reboot the system.
12. When a prompt displays on the terminal's screen, log in then set the TERM variable. To set the variable for one login session, use the `stty` command. To set a default for every login session, set the variable in your `.profile` file.
13. Use the `stty`(1) command to set the baud rate of the terminal to match the rate specified in Step 7. Set the other parameters for the terminal as desired. See the `stty`(1) command in the *User's Reference Manual* for information about setting a terminal's parameters.

Configuring Modems

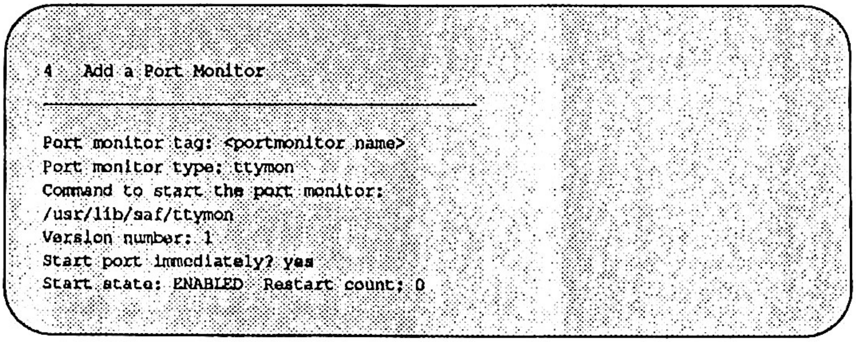
Before connecting a modem to a serial port, you must create a port monitor and port service. You can use the sysadm interface or shell commands. Both methods are described.

Creating a Port Monitor and Port Service using sysadm

To create a port monitor and port service using sysadm menus, do the following:

1. Log in as root.
2. Invoke sysadm.
3. Select the Port Access Services and Monitors menu.
4. Select the Port Monitor Management menu.
5. Select the Add a Port Monitor menu.

   The following form displays:

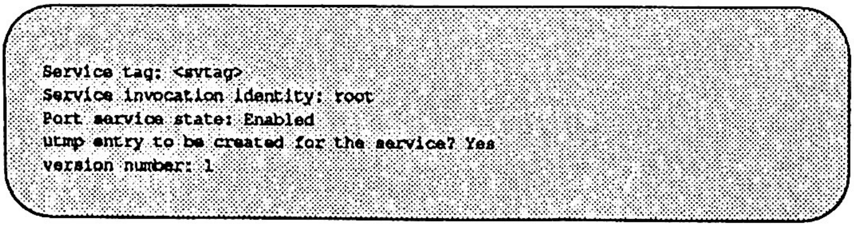


```
4  Add a Port Monitor

Port monitor tag: <portmonitor name>
Port monitor type: ttymon
Command to start the port monitor:
/usr/lib/saf/ttymon
Version number: 1
Start port immediately? yes
Start state: ENABLED  Restart count: 0
```

6. Specify the port monitor name, port monitor type, and the restart count, where:
   - the port monitor tag is the alphanumeric name you assign to the port monitor.
   - the port monitor type is `ttymon`.
   - the restart count is the number of times the `sac` program should try to restart the port monitor if it fails (the default is 0; change it if desired).
7. Save the input by pressing F3.
8. Now you must create a serial port service. To do this, press F6 twice to get to the `Service Access Management` menu.
9. Select the `Port Service Management` menu.
10. Select the `Add Port Services` menu.
11. Select the `Add a service to a particular port monitor` menu.
12. Select the port monitor tag you specified earlier.

The `Add Port Services to Port Monitor <portmon>` form displays. The form looks like this:



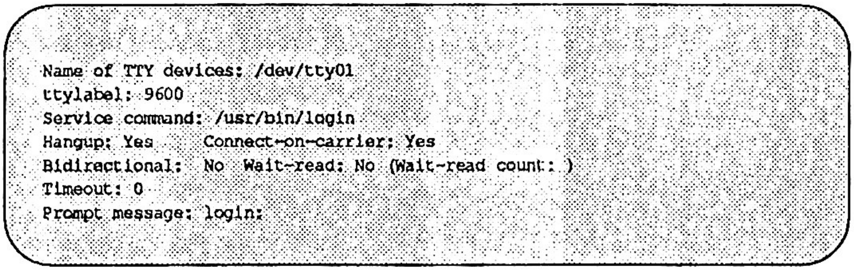
```
Service tag: <svtag>
Service invocation identity: root
Port service state: Enabled
utmp entry to be created for the service? Yes
version number: 1
```

13. Specify the service tag and the service invocation identity, where:

    - the service tag is the name you assign to the service,
    - the service invocation identity is the login name assigned to the service when the service starts. The login id specified must be in the `/etc/passwd` file.
    - The `utmp` entry must be set to `yes` to enable a login through a terminal or modem. The default is `no`.

14. Save the input by pressing F3.

15. The `Add Port Services for ttymon` form displays. Make sure it is filled out as follows (some of the fields are already filled in):



```
Name of TTY devices: /dev/tty01
ttylabel: 9600
Service command: /usr/bin/login
Hangup: Yes     Connect-on-carrier: Yes
Bidirectional:  No  Wait-read: No (Wait-read count: )
Timeout: 0
Prompt message: login:
```

NOTE

Change the Bidirectional field to `Yes` if you need dial-in access.

16. Press F3 to save the data.

17. Exit the `sysadm` interface by pressing F7 e ↵.

18. Connect the modem to the serial port.

19. Shutdown and reboot the system by typing:

    ```
    shutdown -y -g0
    ```
    Ctrl-Alt-Del

    When the system prompt returns, the modem can be used.

Creating a Port Monitor Using Shell Commands

Use the sacadm (1M) command to create a port monitor. The format of the command is:

```
sacadm -a -p <pmtag> -t <type> -c "<cmd>" -v ‘pmspecific -V’ [-f dx] [-n count]
```

where:

- -a means add a port monitor
- -p *pmtag* is the name you assign to identify the port monitor. You can use any alphanumeric name.
- -t *type* is the type of port monitor being added (The choices are ttymon or listen. ttymon is used for terminals and modems.
- -c *cmd* is the command string to execute to start the port monitor. Always use the command /usr/lib/saf/ttymon.
- -v is the version number of the port monitor, which defines the format of its administrative file. Specified as -v ‘*pmspecific* -V’ where *pmspecific* is ttyadm when setting up a port monitor or nlsadmin when setting up a network listener.
- -f dx is the flag, where d means do not enable the new monitor, and x means do not start the new port monitor.
- -n *count* specifies the number of times the sac program should attempt to restart the port monitor if a monitor fails. The default is 0.

NOTE

In the above command, you must use a single back quote character (‘), not a forward quote (’), or the command will fail.

A backslash (\) indicates that the command continues on the next line. You do not have to type the backslash. However, if you type \ then ↵, system prompts you to continue the command by displaying a right angle bracket (>).

For example, to add a ttymon port monitor called portmon, type:

```
sacadm -a -p portmon -t ttymon -c /usr/lib/saf/ttymon -v ‘ttyadm -V’
```

To verify that the new port monitor exists, type:

```
sacadm -l -p portmon
```

The output should look like this:

```
PMTAG   PMTYPE FLGS RCNT STATUS    COMMAND
portmon ttymon -    0    ENABLED  /usr/lib/saf/ttymon #
```

NOTE

The STATUS field may be set to STARTING, instead of ENABLED

Next, you must create a port service.

Creating a Port Service Using Shell Commands

Create a port service using the pmadm(1M) command. This command has the following format:

pmadm -a -p *pmtag* -s *svtag* -i *id* [-f ux] -v *ver* -m *pmspecific*

where:

- -a means add a service
- -p *pmtag* is the name you assigned to identify the port monitor in the previous procedure.
- -s *svtag* is the name you assign to the service. The service tag is part of the entry for the service in the port monitor's administrative file. The svtag can be any 1 to 14 character alphanumeric name you desire.
- -i *id* is the login name (usually root) assigned to the service when it is started. The id must be in the /etc/passwd file. The id checks for access privilege.

- -f ux is a flag that tells the service to either create a utmp entry for the service (u) or to bring up the service in a disabled state (x).
- -v ver is the version number of the port monitor's administrative file. Specified as -v `ttyadm -V` when setting up a port monitor or -v `nlsadmin -V` when setting up a network listener.
- -m is used to specify the port monitor's administrative command.
- -s *service* is the full pathname of the service to be invoked

NOTE

Use a single back quote character (`) where shown, or the command will fail.

Example of Creating a Port Service

For example, to create a port service for tty01 at 9600 baud, with a service tag of 22, do the following:

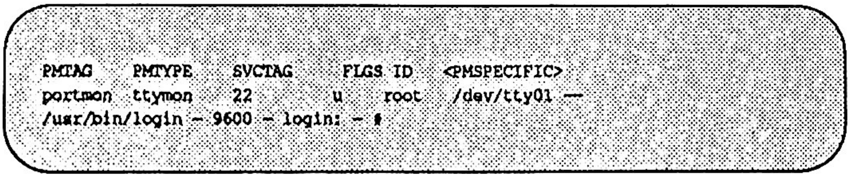
1. Log in as root.
2. Type:

```
pmadm -a -p portmon -s 22 -i root -f u -v `ttyadm -V` -m\
"`ttyadm -d /dev/tty01 -l 9600 -s /usr/bin/login`"
```

3. Verify the entry by typing:

```
pmadm -l -p portmon
```

The output should look like this:



```
PMTAG    PMTYPE    SVCTAG     FLGS ID    <PMSPECIFIC>
portmon  ttymon    22          u   root   /dev/tty01 —
/usr/bin/login - 9600 - login: - #
```

NOTE The FLGS field reports whether the port is enabled or disabled. If an x is in the field, the port is disabled.

4. Enable the ttymon service by typing something like:

```
pmadm -e -p portmon -s 22
```

5. Connect the modem to the serial port.

6. Shut down and reboot the system by typing:

```
shutdown -y -g0
```

Ctrl-Alt-Del

When the system prompt returns, you can use the device attached to the serial port.

NOTE For more information about the pmadm(1M) or sacadm(1M) commands, see the *System Administrator's Reference Manual*.

Reconfiguring a Port for a Modem

If a port was configured for a terminal using the Quick Terminal Setup feature, it is easy to reconfigure the port for a modem.

CAUTION If you want to connect a modem to the serial port but the port was previously configured for a terminal, do not use the Remove a Terminal from a Port menu to remove the terminal. If you do, the entries in the /etc/conf/node.d/asy file will be removed and you will have to reconfigure the serial port. Use the method below to reconfigure the port for a modem.

To reconfigure a serial port for a modem, do the following:

1. Invoke sysadm.

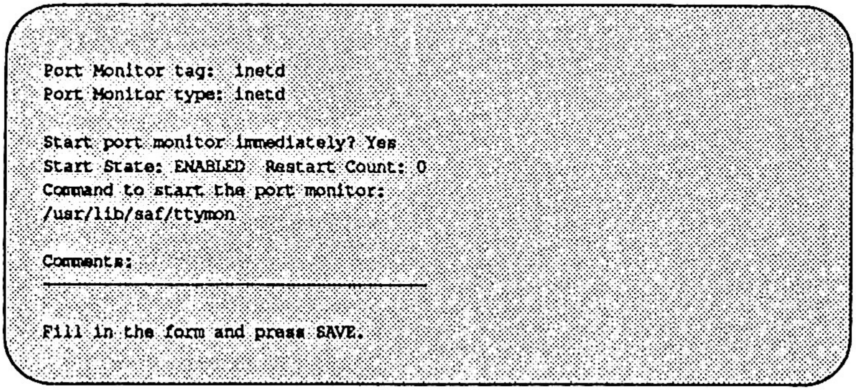
2. Select the `Port Access Services and Monitors` menu.
3. Select the `Port Monitor Management` menu.
4. Select the `Modify a Port` menu.

   The system displays the existing port monitor tags and types.

5. Move the cursor to the port monitor tag you want to modify, then press ↵.

   The `Modify a Monitor` form displays.

6. The form will look something like the following:



```
Port Monitor tag:  inetd
Port Monitor type: inetd

Start port monitor immediately? Yes
Start State: ENABLED  Restart Count: 0
Command to start the port monitor:
/usr/lib/saf/ttymon

Comments:
_______________________________________

Fill in the form and press SAVE.
```

7. Change the desired fields by moving the cursor to a field then pressing the CHOICES key or by typing the information. When the desired information displays in the field, move on to the next field you want to change. For example, to modify the port for a modem instead of a terminal, change the field `Command to start the port monitor` to `/usr/sbin/inetd`.
8. When done, press F3 to save the changes.
9. Exit the `sysadm` interface by pressing F7 ● ↵.
10. Connect the modem to the port.

Configuring Postscript Printers

This section tells you how to directly connect a postscript printer to a computer running SVR4 UNIX, then how to configure the system so remote systems can access the printer. The shell commands and the `sysadm` menus used to do the procedures are both described. Choose the method you desire. For more background information, see Chapter 6 in the *System Administrator's Guide*.

NOTE Postscript printers can only be used on serial ports.

Determining the Printer Configuration

Before you add the printer to the `lp` service, figure out how the printer should be configured. Do the following:

1. Decide on a name for the printer. The name may be one through 14 alphanumeric characters. It is best if the name indicates something about the printer; such as the manufacture's name or whether the printer is postscript. The example that follows will use the name `nec1`.

2. Determine the printer type. The printer type is either PS or PSR. PS is for postscript printers that collate, PSR is for postscript printers that don't collate. The printer type, initializes the printer before a request is sent to it, and is used for filtering different types of input into postscript.

3. Determine the content type of the files that will be printed. By default, the `lp` service assumes that the only kind of file that can be printed without filtering is an ascii file (content type = `simple`). The content type of a postscript printer must be designated as `PS`.

4. Determine which interface program to use. When a serial printer is added, the system (by default) assigns the `standard` interface program to the printer. The `standard` interface program is the only program available for SVR4 Versions 1, 2, and 3.

   The `standard` interface program defines the stty settings (low level communications) used by the printer. This interface program will work with postscript printers, however, there may be some problems. For example, you may occasionally get printer faults. The printer faults seem to be due to timing problems with the `lp` spooler.

NOTE

In SVR4 Version 4 and beyond, you should use the `PS` interface program. If you use `sysadm` to set up the postscript printer, you can't specify an interface file other than the default. So, if you use `sysadm` to configure the printer, you must then use the `lpadmin` command to specify the `PS` interface file. This step is described in more detail later.

5. Look at the manual which comes with your printer to find out which `stty` settings are required. If they are different than the defaults shown below, the settings must be changed. The default settings are reasonable for most postscript printers, so, only change the defaults when required by your printer.

CAUTION

The printer must be in the "interactive" postscript mode, not the "batch" postscript mode.

The default `stty` settings with the `standard` interface program are:

- 9600 (baud rate)
- cs8 (8-bit bytes)
- cstopb (1 stop bit per byte)
- parenb (no parity)
- ixon (XON/XOFF flow control)
- onlcr (map linefeed into carriage return/linefeed)

NOTE

To change the default settings, use the `sysadm` interface and select the `printers` and `communications` menus.

Once you've figured out the desired configuration, you are ready to connect the printer and add the printer to the `lp` service. You can either use the `lpadmin` command or the `sysadm` menus to add and configure the printer.

Connecting the Printer

To connect the printer, do the following:

1. Connect one end of the cable to the printer's serial port.
2. Connect the other end of the cable to the system's serial port.

NOTE **The device name associated with this connection is `/dev/tty01`. A different serial port and device name can be used, if desired.**

3. Plug in the printer's power cable and turn on the power to the printer.

Adding a Printer Using Shell Commands

Use the `lpadmin`(1M) command to add a printer. The format of the `lpadmin` command is:

```
lpadmin -p <printername> -v <device path> -T <printer type> -I <content type> -i <interface file>
```

For example, to add the printer named `nec1` connected to the serial port using the `lpadmin`(1M) command, do the following:

1. Log in as root.
2. Execute one of the next two commands, depending on which version of SVR4 you have:

   a. If you have SVR4 Version 3 (or earlier), type something like:

   ```
   lpadmin -p nec1 -v /dev/tty01 -T PS -I PS
   ```

   This specifies the printer name as `nec1`, device name `/dev/tty01`, printer type `PS`, and content type `PS`. The default `standard` interface file will be used. You can choose any name for the printer.

b. If you have SVR4 Version 4 or beyond, type something like this:

```
lpadmin -p necl -v /dev/tty01 -T PS -I PS -i /usr/lib/lp/model/PS
```

where necl, /dev/tty01, and PS are the printer name, device name, printer type, content type and interface file.

3. If the printer is going to be the default printer, execute the lpadmin command again with the -d option. For example, since the printer called necl will be the default printer, type:

```
lpadmin -d necl
```

4. Tell the lp service to accept print requests by typing:

```
accept necl
```

5. Enable the printer by typing:

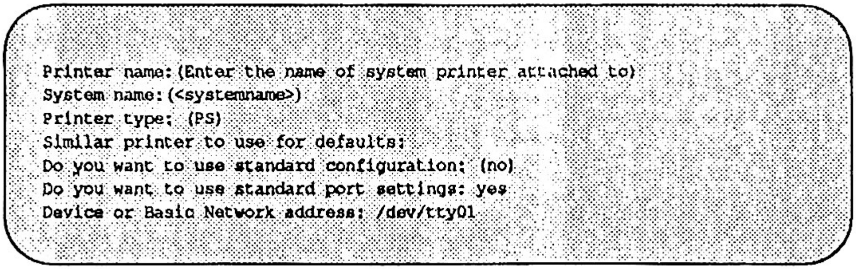
```
enable necl
```

Adding a Printer Using the sysadm Menus

To add a printer to the lp service using the sysadm menus, do the following:

1. Invoke sysadm.
2. Select the following menus: printers, printers, add.

   The following form displays (default fields are already filled. The input required of you is shown in parentheses).



```
Printer name:(Enter the name of system printer attached to)
System name:(<systemname>)
Printer type: (PS)
Similar printer to use for defaults:
Do you want to use standard configuration: (no)
Do you want to use standard port settings: yes
Device or Basic Network address: /dev/tty01
```

3. Enter the desired information, then press F3 to save it.

4. Another form that shows the current printer settings displays. On the line that says "`File types printable without filtering`", change "`simple`" to "`PS`".

5. Press F3 to save the information.

NOTE If you get an error message about using the –v option, press F6 (CANCEL) to go back a menu level, then F3 again. If you still get the error message, exit the `sysadm` interface and use the `lpadmin` command instead.

6. Tell the system that the printer will be the default printer by selecting the following menus: `operations`, `set defaults`.

7. Specify the printer name, then press F3.

8. Tell the `lp` service to accept print requests by selecting the `accept` menu, then pressing F3.

9. Enable the printer by selecting the `enable` menu, then pressing F3.

10. Exit the `sysadm` interface by typing F7 e ↵.

11. If you have SVR4 Version 4 or beyond, you will now have to execute the `lpadmin` command to specify the `PS` interface program, rather than the default `standard` program. To do this, type something like:

```
lpadmin -p nec1 -i /usr/lib/lp/model/PS
```

Testing the Printer

To test the printer, do the following:

1. Type something like:

   ```
   lp -d necl /etc/passwd
   ```

   The contents of the `/etc/passwd` file should print.

   NOTE If you designated the printer as the default printer, you do not have to use the `-d` option.

2. To print an ascii file, type:

   lp *<filename>*

3. To print a troff file that uses the mm macros, you would type:

   troff -mm *<filename>* | lp -T troff

4. If the file does not print, see the "Troubleshooting" section for more information.

5. To find out the status of a print job and the printer(s), type:

   ```
   lpstat -t
   ```

Troubleshooting

If a job doesn't print, do the following:

1. Make sure the power cables and connections are correct and intact.
2. Check the serial configuration. Make sure the `stty` settings on the printer match the `stty` settings on the system and correct any inconsistencies.
3. Make sure that all directories and files under the `/var/spool/lp` directory are owned by `lp`.

4. Make sure the `lp` scheduler is running (execute `/usr/lib/lpsched`)

5. If a job does not print, read the mail sent to the user `lp` to see if there are any error messages.

   NOTE If you are on a single-user system, you may want to have the mail sent to your mail account instead of the `lp` account. To do this, type something like `lpadmin -p necl -A mail` *<username>*.

6. Make sure the `postio` filter looks like the following (type `lpfilter -l -f postio` to display the desired information):

```
Input types: postdown
Output types: PS
Printer types: PS
Printers: any
Filter type: fast
command: /usr/lib/lp/postscript/postio -q
Options: printer * = -L/dev/null
```

7. If the file does not look like the above example, create the file, then type:

   `lpfilter -f postio -F` *<name of file>*

8. Shut down and restart the `lp` service by typing:

```
lpshut
/usr/lib/lpsched
```

NOTE For more information, see Chapter 6 in the *System Administrator's Guide*.

Configuring the System For Network Printer Access

This section tells you how to configure the `lp` system so a postscript printer can be accessed over a network. You must be logged in as root to do the procedures.

Configuring the Server System

To configure a system for access to its printer over a network, do the following:

1. Directly connect the printer to the system and configure it as described in the previous procedures.
2. While logged in as root, type:

   `lpsystem` *<systemname>*

   where *systemname* is the name of another system that has permission to access the printer over the network.

Configuring the Client System

Next, configure the print service on the client system (the system that wants to send a print job over the network), as follows:

1. Log in as root on the client system.
2. Specify the name of the server system by typing:

   `lpsystem` *<systemname>*

3. Tell the `lp` system the content type of the files you plan to send to the printer. When printed, files with the specified content type will be filtered on the server system, not the client. For example, if postscript, troff, and ascii files will be printed on a printer called `qms` on the system called `inbox`, type:

   ```
   lpadmin -p qms -T PS -I postscript,troff,simple -s inbox!qms
   ```

4. If the printer is going to be the default printer, type:

   ```
   lpadmin -d qms
   ```

5. To activate this change, shut down and restart the print service by typing:

```
lpshut
/usr/lib/lpsched
```

6. Type the following:

```
accept qms
enable qms
```

7. You should now be able to send a print job over the network. For example, to send a file that contains tables and is formatted with `troff` and the mm macros to the printer on a remote system, type:

```
tbl <filename> | troff -mm | lp -d qms -T troff
```

where qms is the printer name.

NOTE

If you did Step 4, you don't have to type "`-d` *<printername>*" in the above command. The system automatically sends the job to the default printer.

The `tbl` and `troff` utilities are in the `/usr/ucb` directory. This directory is not in your default PATH.

See the `lp`(1) command in the *User's Reference Manual,* and the `lpadmin`(1M) command in the *System Administrator's Reference Manual* for more information, if desired. Also see Chapter 6 in the *System Administrator's Guide* for more information about printer configuration and maintenance.

4 X WINDOW SYSTEM

Introduction

This chapter discusses topics related to using the X Window System. The topics in this chapter are:

- Configuring more windows
- Using X over a TCP/IP network
- Configuring a system to be an X Window server
- Running X clients on another host.

Increasing Allowable X Windows

The number of windows that can be open on a system depends on the number of stream buffers allocated when the nsu package was installed, and the amount of memory on your system. The default maximum number of xterm windows allowed is 5. You can increase the number of xterm windows by tuning some system parameters and by altering values in some system files.

Tune system parameters either by editing the /etc/conf/cf.d/stune file and changing the value of the desired parameter, or by executing the idtune(1M) command (which edits the /etc/conf/cf.d/stune file).

NOTE Appendix B in the *System Administrator's Guide* describes tunable parameters in detail.

To increase the number of allowable X windows, do the following:

1. Log in as root.
2. Increase the number of stream buffers by typing:

   ```
   /etc/conf/bin/idtune NSTREAM 128
   ```

   The system responds:

   ```
   Tunable Parameter NSTREAM is currently set to xxx.
   Is it ok to change it to 128? (y/n)
   ```

3. Type y.

   The NSTREAM parameter in the stune file is changed.

4. Increase the number of processes that can be invoked at one time by typing:

   ```
   /etc/conf/bin/idtune MAXUP 60
   ```

   The MAXUP value in the stune file is changed.

5. Edit the /etc/conf/cf.d/mdevice file,. Find the line for consem and change the value in the eighth (next-to-last) column to 128 (the default is 64).

6. Change directory to /etc/conf/sdevice.d.

7. Edit the ptm file and change the value in the third column to 64 (the default is 16).

8. Edit the ptem file and change the value in the third column to 64 (the default is 16).

9. Edit the consem file and change the value in the third column to 64 (the default is 16).

10. By default, SVR4 provides 16 pts (pseudo-terminal) devices. When the nsu package was installed, the Installation Guide recommended that you specify at least 64 devices. (See Appendix C in the *System V Release 4.0 Installation Guide* for details.) If you only installed the default number of pts devices, do the following; otherwise, skip this step:

    NOTE If you do not know how many pts devices are already configured, type ls /dev. Count how many entries begin with pts.

    To increase the number of pts devices, first make a backup copy of the /etc/conf/node.d/pts file, then type the following:

```
for i in 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30\
31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49\
50 51 52 53 54 55 56 57 58 59 60 61 62 63
  do
  echo "pts<Tab>pts/$i<Tab>c<Tab>$i" >>\
    /etc/conf/node.d/pts
  done
```

NOTE Do not type the backslashes. They are just to show you that the command line continues.

This command puts a line for each new `pts` device into the configuration file so the devices are created when the kernel is rebuilt.

11. Rebuild the kernel by typing:

```
/etc/conf/bin/idbuild
```

12. When the # prompt returns, shut down and reboot the system by typing:

```
shutdown -y -g0
```

Ctrl-Alt-Del

When the system prompt returns, you can log in and access more windows.

NOTE If you have 25 windows open at once, system performance is greatly reduced.

Using X Over a TCP/IP Network

If X Windows and TCP/IP is installed on your system, you can execute any of the windowing commands on one system and have the output displayed on another system. The X command that is executed is called the X client. The system displaying the output is called the X server (because it is providing display services to the client).

NOTE To find out what X commands you can use, type `ls /usr/X/bin`. For details about these commands, see the *OPEN LOOK Graphical User Interface User's Guide*. Appendix A contains the xterm man pages.

To use X Windows over the network, both systems must:

- be running the same version of the X Window System
- know each other's host name (the host name of each system must be in the other system's `/etc/hosts` file).
- be running System V Release 4 (SVR4).
- be running TCP/IP (the `inet` package).

NOTE To find out what packages (and their version numbers) are installed on your system, type `pkginfo`.

There are two ways you can initialize X. You can initialize the default X environment by typing `xinit`, or you can initialize OPEN LOOK. To initialize OPEN LOOK, first add yourself as an OPEN LOOK user by typing `/usr/X/adm/oladduser`, then type `olinit` to start the default OPENLOOK environment). See the *OPEN LOOK Graphical User Interface User's Guide* for information about how to customize your OPEN LOOK environment.

Configuring a System to be an X Window Server

You can temporarily or permanently permit another host to use your system as an X Window server.

To see which hosts already have permission to access the X Window System server on your machine, type:

```
/usr/X/bin/xhost
```

A list of hosts that have access permission displays. If the desired host does not already have permission, do one of the following procedures.

To temporarily permit a host to use your machine as an X Window System server, do the following:

1. Log in as root on the server system.
2. Type:

   `/usr/X/bin/xhost +` *hostname*

   This adds *hostname* to the list of hosts that are allowed to access the X Window System server on your machine.
3. Log in as root on the client system.
4. Repeat Step 2.

   The permission is valid until you exit and restart the X Window System.

To permanently add the host, do the following:

1. Log in as root on the server system.
2. Edit the `/etc/x0.hosts` file and add the name of the desired host.
3. Log in as root on the client system.
4. Repeat Step 2.

   Now you can start an X client on another host with the output displayed on the X server, as described next.

Running X Clients on Another Host

Before you execute a windowing command on another system, keep the following in mind:

- Most windowing commands are in the `/usr/X/bin` directory. This directory is not in the default PATH in your `.profile` file.
- If you add the X directory to the PATH variable in your `.profile` file, note that the `rsh` command does not read the `.profile` file. However, the `rlogin` and `rcp` commands do read it. So, when using the `rsh` command to execute a windowing command on a remote host, you must specify the full path of the X command.
- X must be actively running on the remote system.

To start an X client process on a remote host with the display on the local host, do the following:

1. Verify that you can execute a command on the remote system by typing:

   ```
   rsh hostname ls -al
   ```

   The system should respond with a long listing of your home directory on the other machine.

   NOTE

   You can't be logged in as root when executing this command.

   If the system reponds: `<hostname>: cannot open`, the remote host may only allow a few `rlogins` at a time and the limit has already been reached. Try to execute the command later, or ask the system administrator to increase the number of `rlogins`, if possible. The multi-user license package must be installed on the system to get more than two rlogins.

   This error message also appears if you are executing the command from a system running a pre-SVR4 version of UNIX.

2. To execute windowing commands on a remote host, use the following syntax:

   ```
   rsh remotehost: /usr/X/bin/xclient -display localhost:0&
   ```

   where *remotehost* and *localhost* are the two systems and *xclient* is the program you want to run over the network.

For example, to start an xterm client process on the remote host (wizard) with the output displayed on the local host (gollum), type:

```
rsh wizard /usr/X/bin/xterm -display gollum:0&
```

The xterm command is executed on the system called wizard. If wizard has a .Xdefaults file, the file is searched to see which xterm variables to use to create the xterm window. Otherwise, the xterm defaults are used.

NOTE There is a limit to how many windows can be open at one time. If the command fails to execute, it may be because the limit has been reached. By default, you can only open 5 xterm windows at a time. This limit can be increased by doing the procedure called "Increasing Allowable X Windows".

5 NETWORKING

Introduction

This chapter describes how to set up and use features that work with TCP/IP networking. The topics covered include:

- Establishing/controlling network access
- file transfer protocol (`ftp`)
- Troubleshooting

If your system does not have TCP/IP networking software (the `inet` package) installed, this chapter is not for you. To find out if your system has this software, use the `pkginfo` command and look for the `inet` package, or type:

```
ping localhost
```

If the message `localhost is alive` displays, it means that TCP/IP is installed on your system.

Establishing/Controlling Network Access

This section describes how to set up your network so users on remote systems can remotely log in (rlogin), remotely copy files to/from your system (rcp), or remotely execute commands on your system (rsh). Also described are some things you can do to control who has access to your system and how much access they have. Access to a system can be controlled at the system level and at the user level. The following procedures describe both.

Controlling System Level Access

System-level access can be controlled by either specifying system-wide equivalent hosts, or by completely disabling remote access, as described next.

System-wide Equivalent Hosts

An equivalent host is a machine from which other users can use the rlogin, rcp or rsh commands to access your system. You can permit all hosts or specific hosts in your local area network to be equivalent hosts. You can also completely disable remote access to your system.

Equivalent hosts are established by creating the /etc/hosts.equiv file. Only root can create and edit this file.

NOTE

The sysadm interface cannot be used to establish equivalent hosts.

Specifying Equivalent Hosts

To specify equivalent hosts, do the following:

1. Log in as root.
2. Use your favorite editor to create the /etc/hosts.equiv file.
3. If you trust all the hosts in your network (and all the users on the hosts), put a + on a line by itself.
4. To specify specific hosts that you trust, enter the name of the host. For example, if you add the following entries:

```
gandalf
smeagle
smirnoff
```

all users who can log into the hosts gandalf, smeagle, and smirnoff will be able to rlogin to your system (or execute rcp or rsh). If the user trying to access your system has an entry in your system's /etc/passwd file, they can log in without supplying a password; otherwise, they will be asked for a password.

NOTE

System-wide equivalent hosts are a potential security problem. If you only want certain users on a remote host to access your system, do not put the host name in the /etc/hosts.equiv file. See the section " Controlling User-Level Access" for details.

Disabling Remote Access

If desired, you can completely disable rlogin, rpc, or rsh access to your system using shell commands or the sysadm interface.

To use shell commands to disable remote access, do the following:

1. Log in as root.
2. Edit the /etc/inet/inetd.conf file.

   The file, in part, looks like the following:

   ```
   # Shell, login, exec, comsat and talk are BSD protocols.
   #
   shell   stream tcp nowait root /usr/sbin/in.rshd     in.rshd
   login   stream tcp nowait root /usr/sbin/in.rlogind in.rlogind
   exec    stream tcp nowait root /usr/sbin/in.rexecd   in.rexecd
   comsat dgram  udp wait   root /usr/sbin/in.comsat   in.comsat
   talk    dgram  udp wait   root /usr/sbin/in.talkd    in.talkd
   ```

3. Comment out (type a # at the beginning of) the line that contains the function you want to disable. For example, to disable rlogin capability, comment out the 4th line shown, which contains:

   ```
   /usr/sbin/in.rlogind    in.rlogind
   ```

4. Save the changes you made to the file and exit the editor.
5. Determine the *inetd process id* that is currently running by typing:

   ```
   ps -e | grep inetd
   ```

   The number in the far left column is the process id number.
6. Signal the `inetd` daemon to reread the `inetd.conf` file by typing:

   `kill -Hup` <*inetd process id*>

   Now, anyone who attempts to use the disabled function will get a `Permission denied` message.

To disable all remote access using the `sysadm` interface, do the following:

1. Log in as root.
2. Type `sysadm`.
3. Select the `ports` menu.
4. Select the `Port Monitor Management` menu.
5. Select the `disable` menu.
6. Select `inetd` by pressing F2, then F3.
7. Press F3 to save the data.
8. Exit `sysadm` by pressing F7 e ↵.

NOTE To disable `rcp` access, make sure the host you want to disable is not in the `/etc/hosts.equiv` file on your system.

Disabled Functions You May Want to Enable

There are several functions that are disabled by default. You can enable them, if you want to, as described in this section.

The finger Command

The /etc/inet/inetd.conf file contains an entry for the finger(1) command. The finger command reports to the requester the full name and user name of everyone who is logged into the local (or remote) system on your network. There is no record of who initiated the request for the information. This may be a security leak in systems that require strict security. So, as a security measure, this function is disabled.

To enable the finger command for your system, do the following:

1. Log in as root.
2. Edit the /etc/inet/inetd.conf file.
3. Uncomment the line that begins #finger.
4. The file is read-only. Force the change to the file. For example, if you are using the vi editor, write the data and quit the file by typing:

   ```
   :x!
   ```

5. Find out the process id of inetd by typing:

   ```
   ps -e | grep inetd
   ```

   The number in the far left column is the process id number.
6. Signal the inetd daemon to reread the inetd.conf file by typing:

   ```
   kill -Hup <inetd process id>
   ```

   The finger command can now be used.

The rwhod Command

By default, the rwhod(1M) command, which also reports who is logged into the system, is not executable.

To enable the rwhod command, do the following:

1. Log in as root.
2. Edit the /etc/inet/rc.inet file.

3. Append the following line to the end of the file:

   ```
   /usr/sbin/in.rwhod
   ```

4. The file is read-only. Force the change to the file. For example, if you are using the vi editor, write the data and quit the file by typing:

   ```
   :x!
   ```

5. Shut down and reboot the system by typing:

   ```
   shutdown -y -g0
   ```

   Ctrl-Alt-Del

   The rwhod command is now enabled.

Controlling User-Level Access

User's who have an account on your system can set up their account so they can rlogin without a password. A user with an account on your system can grant other users permission to log into the system using their login id (this is a potential security problem).

This section describes how to:

- get access to another system without having to specify a password,
- allow other users to use your login,
- restrict user access.

Granting Access Without Passwords

If you have a user account on two systems that are connected on a network (we'll call them systems alpha and beta), and you want to log into each system without having to specify your password, do the following:

1. On system alpha, log in as yourself (not root). Since you already have an account on system alpha, it is presumed that you already have a password on alpha.

2. Create a .rhosts file in your home directory that contains the entry:

   ```
   beta
   ```

   Now, when you rlogin to system alpha from system beta, you will not be prompted for a password.

3. If you want to log in to system beta from system alpha without supplying a password, log into system beta as yourself and repeat Step 2.

Allowing Other Users to Use Your Login

If you have an account on a system, you can allow other users to log into the system using your login identity. They will have access to the same files and directories on the system as you, since they will be assuming your identity.

To give permission for another user to use your login, do the following:

1. Log into the system as yourself. For example, lets say your login name is steve.

2. Get in your home directory. For example, type:

   ```
   cd /home/steve
   ```

3. Edit the .rhosts file and enter the name of other hosts and users on the other hosts who have permission to access the system as you. For example, entries such as:

   ```
   beta susan john
   samson paul
   ```

   means that user susan and john can rlogin to your system from system beta as you, and user paul can rlogin to your system from system samson as you.

NOTE

This a potential security problem. For example, the users john and susan can create a .rhosts file in their home directory and grant other users access to their login account. Those users will be able to log into system alpha as user steve, even if steve did not grant them permission, since they have permission to assume the identities of other users. If this is a concern, see Appendix A in the *System Administrator's Guide* for more information about system security.

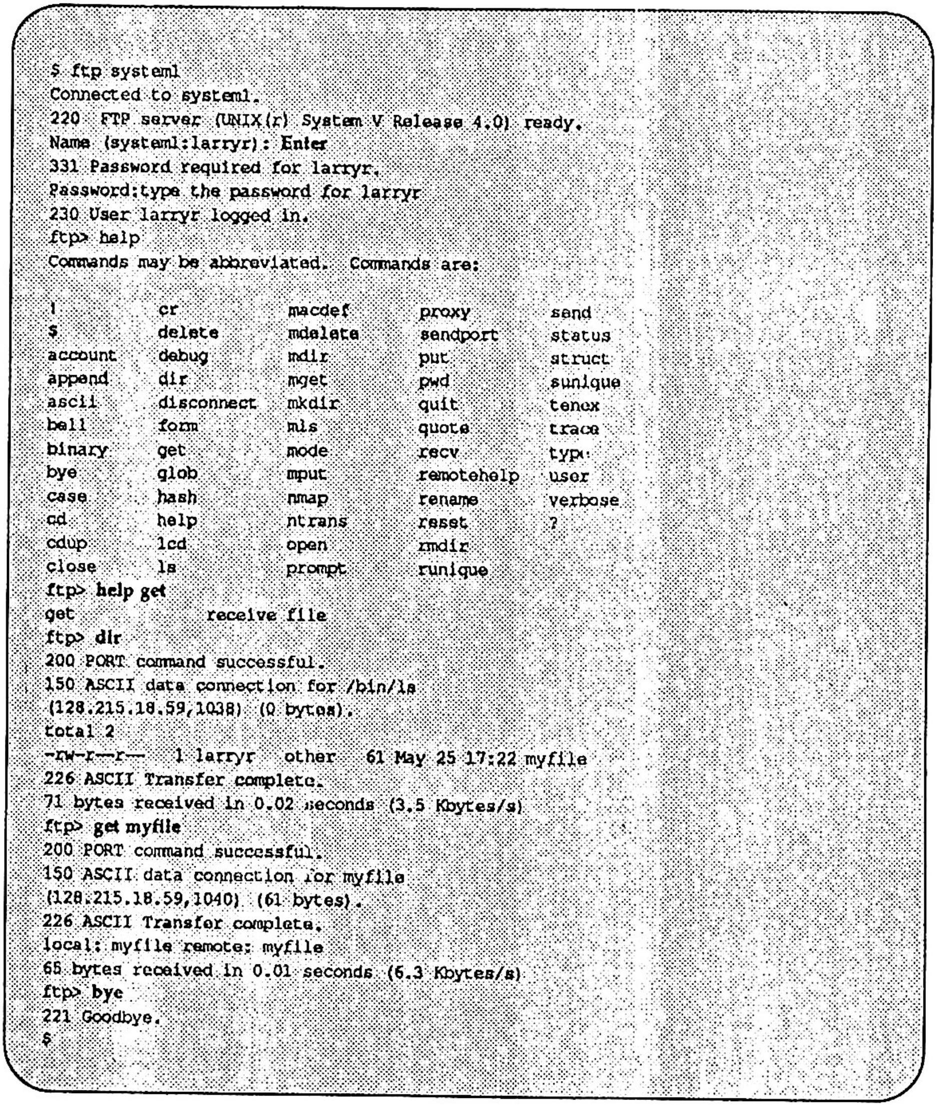
File Transfer Protocol (ftp)

File Transfer Protocol (`ftp`) is used to transfer ASCII and binary files between two machines on a network. `ftp` has many commands as well as an on-screen help. To use `ftp`, you must have or know a login on the machine that you wish to access (or the machine must support anonymous `ftp`).

An Example ftp Session

The following example illustrates a simple file transfer between two systems using `ftp`. In network terms, a local machine is one at which you type commands; a remote machine is one that you access through a local machine across a network.

The example starts out at the system prompt on the local machine. All of the bold words are typed at the local machine; other lines are output from the `ftp` program.



```
$ ftp system1
Connected to system1.
220 FTP server (UNIX(r) System V Release 4.0) ready.
Name (system1:larryr): Enter
331 Password required for larryr.
Password:type the password for larryr
230 User larryr logged in.
ftp> help
Commands may be abbreviated.  Commands are:

!          cr           macdef     proxy         send
$          delete       mdelete    sendport      status
account    debug        mdir       put           struct
append     dir          mget       pwd           sunique
ascii      disconnect   mkdir      quit          tenex
bell       form         mls        quote         trace
binary     get          mode       recv          type
bye        glob         mput       remotehelp    user
case       hash         nmap       rename        verbose
cd         help         ntrans     reset         ?
cdup       lcd          open       rmdir
close      ls           prompt     runique
ftp> help get
get               receive file
ftp> dir
200 PORT command successful.
150 ASCII data connection for /bin/ls
(128.215.18.59,1038) (0 bytes).
total 2
-rw-r--r--   1 larryr   other   61 May 25 17:22 myfile
226 ASCII Transfer complete.
71 bytes received in 0.02 seconds (3.5 Kbytes/s)
ftp> get myfile
200 PORT command successful.
150 ASCII data connection for myfile
(128.215.18.59,1040) (61 bytes).
226 ASCII Transfer complete.
local: myfile remote: myfile
65 bytes received in 0.01 seconds (6.3 Kbytes/s)
ftp> bye
221 Goodbye.
$
```

The result of the example is that the file called myfile is copied to the current working directory of the local machine. The file still exists on the remote machine. After logging into the remote system, you are placed in the HOME directory for larryr (/usr/larryr). Note the use of both a general help and a help for a specific command.

Anonymous ftp

In trusted environments such as most office networks, it is often convenient to have a place to put files where everyone can access them easily. Anonymous ftp can be used to provide this open access.

NOTE

Anonymous ftp is inherently insecure and should be used only in trusted environments.

The following example illustrates an anonymous ftp login into a system called server1. The commands you would type are shown in boldface.

```
$ ftp server1
Connected to server1.
220 server1.intel.com FTP server (Version 4.160 #1 Mon
Mar 5 17:38:00 PST 1990)
ready.
Name (server1:larryr): anonymous
331 Guest login ok, send ident as password.
Password:type your real name
230 Guest login ok, access restrictions apply.
ftp> dir
200 PORT command successful.
150 Opening data connection for /bin/ls
(128.215.18.26,1120) (0 bytes). (mode ascii).
total 513
-rw-r—r—    1 root     root   1109 May 22 13:34 README
dr-xr-xr-x   2 root     root     80 Jan 21 15:10  bin
drwxr-xr-x   2 root     root     64 Nov 22  1989  dev
dr-xr-xr-x   2 root     root     96 Feb  5 11:09  etc
drwxrwxr-x   5 root     ftp     672 May 23 12:44  ftp
drwx———      2 root     root     64 Nov 22  1989  shlib
226 Transfer complete.
755 bytes received in 1 seconds (0.737 Kbytes/s)
ftp> get README
200 PORT command successful.
150 Opening data connection for README() (109 bytes).
(mode ascii)
226 Transfer complete.
local: README remote: README
1144 bytes received in 0 seconds (1.12 Kbytes/s)
ftp> bye
221 Goodbye.
$
```

Setting Up an Anonymous ftp Server

CAUTION Set up anonymous `ftp` carefully, as security problems can occur in "hostile" environments.

The following procedure explains the steps needed to set up a system as an anonymous `ftp` server. The procedure assumes that you already have a user with the login ID `ftp`.

1. Log in as root.
2. Create a `bin` directory for executable files in the home directory of user `ftp` by typing:

   ```
   cd /home/ftp
   mkdir bin
   chmod a-rw bin
   ```

3. Copy the executable file called `ls` into the newly created directory and modify its access permissions by typing:

   ```
   cp /usr/bin/ls bin
   chmod 0111 bin/ls
   ```

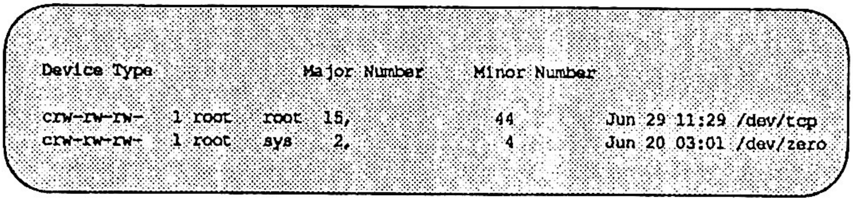
4. Create a directory called `dev` and change the directory's permissions by typing:

   ```
   mkdir dev
   chmod a-rw dev
   ```

5. Determine the major number and minor number of the tcp zero devices as shown in the `/dev` directory by typing:

   ```
   ls -l /dev/tcp /dev/zero
   ```

The system displays something like the following:



```
Device Type               Major Number    Minor Number

crw-rw-rw-   1 root   root  15,              44         Jun 29 11:29 /dev/tcp
crw-rw-rw-   1 root   sys    2,               4         Jun 20 03:01 /dev/zero
```

6. Create new device nodes by typing:

   ```
   mknod dev/tcp c major_number minor_number
   mknod dev/zero c major_number minor_number
   ```

   where *major_number* and *minor_number* are the numbers shown as the output in Step 5.

   For example, using the output of Step 5 for the major and minor numbers, you would type:

   ```
   mknod dev/tcp c 15 44
   mknod dev/zero c 2 4
   ```

7. Create a directory named `etc` and place reduced copies of the files `passwd` and `group` in it. These files should include only those entries needed for the anonymous login (`ftp` and perhaps `root`). Do not include your entire user list as this could create a security risk. The file permissions should be modified.

   Execute the following commands:

   ```
   mkdir etc
   chmod a-rw etc
   cp /etc/passwd etc
   cp /etc/group etc
   cp /etc/netconfig etc
   chmod 0444 etc/passwd etc/group etc/netconfig
   ```

8. Create `usr` and `lib` directories and place a copy of the file `libc.so.1` there by typing:

```
mkdir usr
mkdir usr/lib
chmod a-rw usr usr/lib
cp /usr/lib/libc.so.1 usr/lib
```

9. The anonymous user should have access to a directory in which the publicly available files are kept. So, type the following:

```
su ftp
mkdir pub
chmod 0775 pub
```

The files and directories should have the following permissions and owners.

File or Directory	Permissions	Owner
/home/ftp	dr-x--x--x	ftp
/home/ftp/bin	d--x--x--x	root
/home/ftp/bin/ls	---x--x--x	root
/home/ftp/dev	d--x--x--x	root
/home/ftp/dev/tcp	crw-rw-rw-	root
/home/ftp/dev/zero	crw-r--r--	root
/home/ftp/etc	d--x--x--x	root
/home/ftp/etc/group	-r--r--r--	root
/home/ftp/etc/passwd	-r--r--r--	root
/home/ftp/etc/netconfig	-r--r--r--	root
/home/ftp/pub	drwxrwxrwx	ftp
/home/ftp/usr	d--x--x--x	root
/home/ftp/usr/lib	d--x--x--x	other
/home/ftp/usr/lib/libc.ser±xr-xr-x		other

NOTE

Placing a user name in the file `/etc/ftpusers` denies that user access to the system through `ftp`.

TCP/IP Network Troubleshooting

Local area networks are complicated systems of software and hardware, and there are a lot of ways things can go wrong. Although some problems require the assistance of a system administrator to solve, there are many you can fix yourself. While troubleshooting any network problem, keep in mind that there are three typical points of failure; the local machine, the remote machine, and the network between them.

This section explains the causes and solutions of some common networking error messages. The messages are listed in the order in which they are likely to be encountered.

Troubleshooting Basics

In order to use the resources of a remote host over the network:

- The network cable must be connected to both hosts.
- Your workstation must know the remote host's internet address.
- You must have a valid account on the remote host.
- For some commands, the remote host must be an equivalent host to your workstation.
- You must not be operating as root, or the remote system's `.rhost` file must be set up to allow you root privileges.
- Your workstation must be able to contact the remote host over the network.
- A remote resource must be mounted to your local machine.

The solutions to many networking problems require cooperation between the system administrators of the systems involved. If you are having network problems, you should contact the administrator of the other system. The following sections will help you figure out what to ask that person to do for you.

Unknown Host Messages

If you see the message:

```
hostname: unknown host
```

it means that the specified remote host is not listed in your workstation's /etc/hosts file.

The /etc/hosts file associates host names with internet addresses. A remote host must be listed in this file before you can contact it over the network. For some commands, your workstation must also be listed in the other host's /etc/hosts file; see the discussion of "Permission denied" messages later in this section.

If you see the unknown host message, add the remote hosts name to the /etc/hosts file on your system.

Network is Unreachable Messages

If you see the message:

```
hostname: Network is unreachable
```

it means that the specified remote host is listed in your /etc/hosts file, but is not on the same local area network. Ask your network administrator to establish a gateway between the two networks.

Login Incorrect Messages

If you see the message:

```
Login incorrect.
```

it means that you do not have an account on the specified remote host. Ask the system administrator of the remote host to give you an account. If your account on the remote host is under a different name than on your local host, you must access the remote host by using the `-l` option of the `rlogin` command and your login name on the remote system.

NOTE See the *Network User's and Administrator's Guide* for more information on specifying a different user ID in a networking command.

Permission Denied Messages

If you see the message:

```
Permission denied.
```

when using the command `rcp` or `rsh`, it may be due to one of the following:

- You do not have the necessary read or write permission on the file(s) you are trying to use. (File permissions work the same over the network as they do on a single workstation.)
- You are operating as root. For security reasons, root is not normally allowed to access other workstations over the network.

- Your workstation is not listed in the remote host's `/etc/hosts` file. Ask the system administrator of the remote host to add it for you, or add it yourself.
- Your password on the remote host is invalid or has expired. Use the `rlogin` command to log into the remote host and change your password.
- You have an account on the specified remote host, but your workstation is not an equivalent host to the remote host.

Not on System Console Messages

If you see the message:

```
Not on system console
```

after executing the `rlogin` command, it means that you are operating as root. root is not allowed to access other workstations over the network.

No Messages

If you get no response from an `rlogin`, `rcp`, or `rsh` command, and the command never completes, it means that the remote host or the network is not up and running. First, press Del to abort the command. Then, ask the administrator of the remote host for help, or just wait for the host or network to return to service.

To check whether the network and the remote host are back in service, type:

`/usr/bin/ping` *hostname*

The `ping` command sends out an "are you there?" signal (like a sonar ping) to the named host and waits for a reply.

If the remote host is accessible over the network, the system says, *hostname* `is alive`. Otherwise, the system says, `no answer from` *hostname*. (The `ping` command tries several times to contact the named host, so it may take a while

message to appear.)

The `no answer from` *hostname* message can result from one or more of the following causes:

- The specified remote host is not currently up and running.
- The remote host is up, but its networking software is not running.
- There is a problem with the network (for example, the network cable is unplugged at some point).

If the solution to the problem is not obvious, ask your system administrator for help or see the *System Administrator's Guide*.

NFS Troubleshooting

In general there are three problems that are most common in systems using NFS. These are:

- an overloaded server,
- a table contains a misspelling,
- something is physically disconnected on the network.

In dealing with NFS problems, check for these three common problems first. If no solution is found by checking these, then try to determine where the problem is located. If the problem is confined to a single client, then the problem is probably on that client. If several clients are experiencing problems, then check the server.

Stale File Handle

The message, `Stale File Handle`, is an NFS error that usually occurs when your system opened a file on an NFS server, then after the file was opened, it was replaced on the server. Accessing the file again usually corrects the problem.

For administrators of NFS servers, you should replace a binary file that may be used by clients by doing the following:

1. `mv` the binary to a temporary name.
2. `mv` the new binary to the old name.

Users who were accessing the earlier version will continue to access it without an error message. Future users will receive the new version of the file.

Slow Access Time on an NFS Client

Slow access to an NFS server that is confined to a single client may be caused by a problem with the client's `biod` processes. Do the following:

1. Log in as root on the client system.
2. Get the process ID's for the `biod` daemons by typing:

   ```
   ps -ef | grep biod
   ```

3. Kill the processes by typing:

   `kill -9` *pid1 pid2 pid3 pid4*

4. Restart the `biod` daemons by typing:

   ```
   /usr/lib/nfs/biod 4
   ```

Occasional Slow Access on All NFS Clients

If all the NFS clients are having trouble with access time, find out if some of the users are executing commands that are not appropriate for an NFS client. For example, the `find` command, which is used to locate files and directories, is inappropriate to use using NFS. If a user wants to use the `find` command on a remote system, they should `rlogin` to the server, then execute the `find` command.

Index